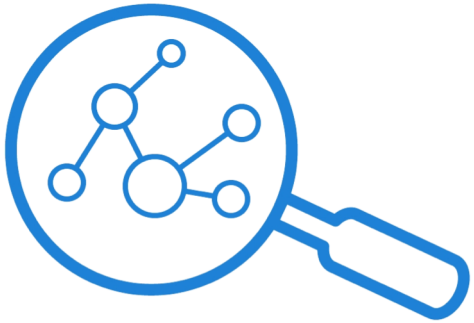
数据科学与大数据技术的数学基础

第一讲



计算机学院
余皓然

2023/4/18

课程内容介绍

课程背景

对海量数据的存储/分析/...的需求日益增长

课程背景

对海量数据的存储/分析/...的需求日益增长

新浪微博至少有2亿日活用户，平均每秒至少有上千条微博产生

课程背景

对海量数据的存储/分析/...的需求日益增长

新浪微博至少有2亿日活用户，平均每秒至少有上千条微博产生

1 如何屏蔽重复内容？假设有用户发了一条微博，如何检测这条微博是否在历史上（比如几天前/几年前）出现过

* 避免通过存储历史上所有微博、对它们进行搜索检测来实现

课程背景

如何屏蔽重复内容？假设有用户发了一条微博，如何检测这条微博是否在历史上（比如几天前/几年前）出现过

课程背景

对海量数据的存储/分析/...的需求日益增长

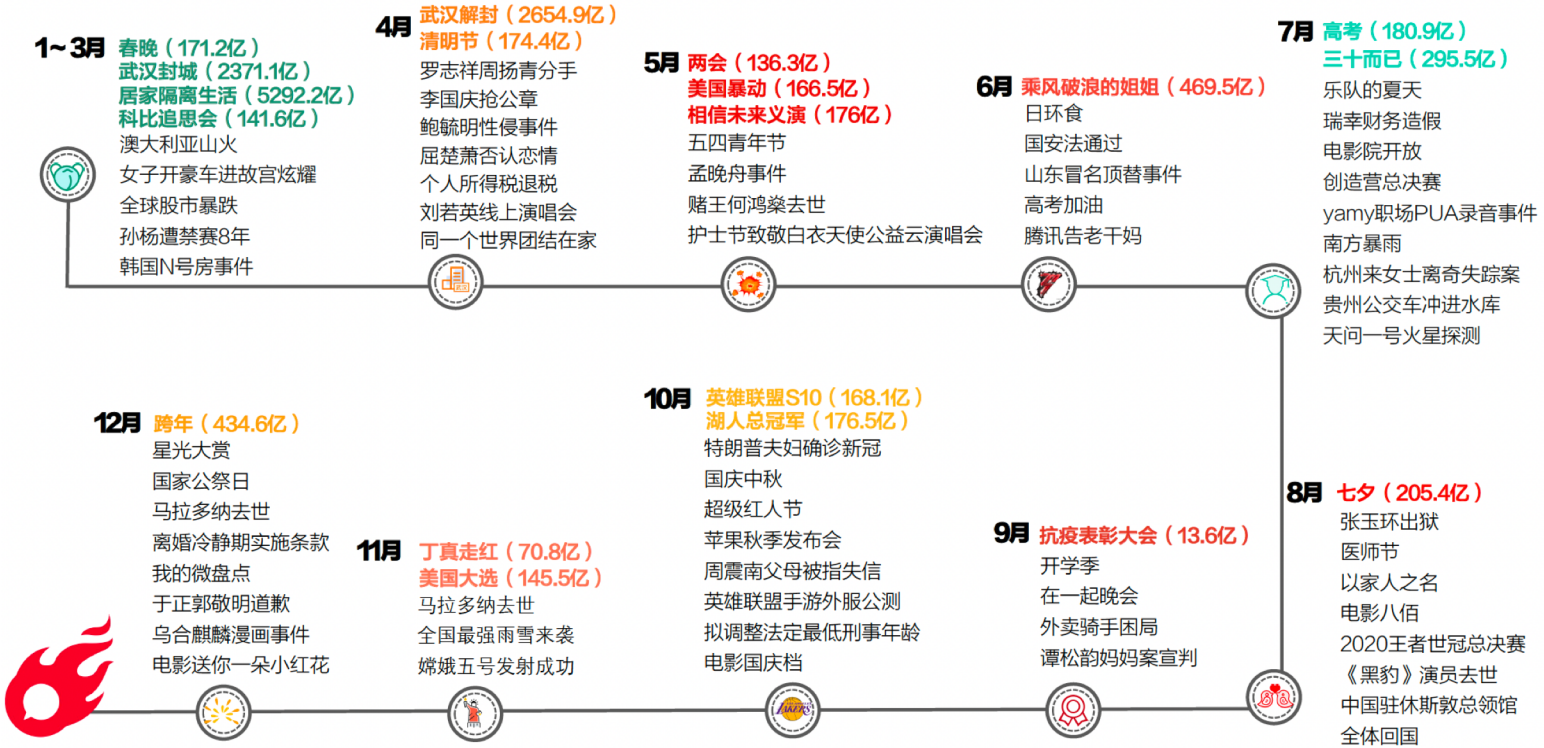
新浪微博至少有2亿日活用户，平均每秒至少有上千条微博产生

2 如何估测在一段时间内各个关键词被搜索/评论的次数

* 避免通过存储这段时间内所有微博、对它们进行搜索检测来实现

课程背景

如何估测在一段时间内各个关键词被搜索/评论的次数



《2020微博用户发展报告》第11页

课程背景

对海量数据的存储/分析/...的需求日益增长

支付宝有超过10亿用户，提供刷脸支付等服务

课程背景

对海量数据的存储/分析/...的需求日益增长

支付宝有超过10亿用户，提供刷脸支付等服务

3 如何迅速完成输入人脸与存储的海量人脸图像间的匹配?

课程背景

对海量数据的存储/分析/...的需求日益增长

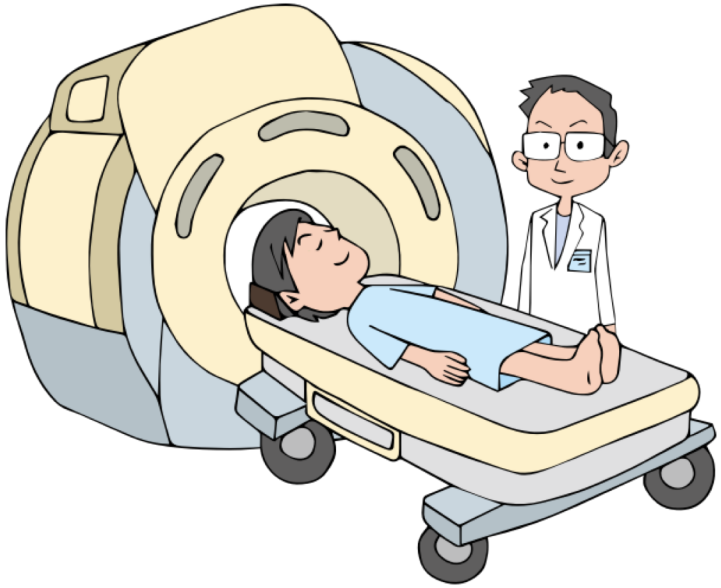
核磁共振成像（MRI）被广泛用于医疗诊断

课程背景

对海量数据的存储/分析/...的需求日益增长

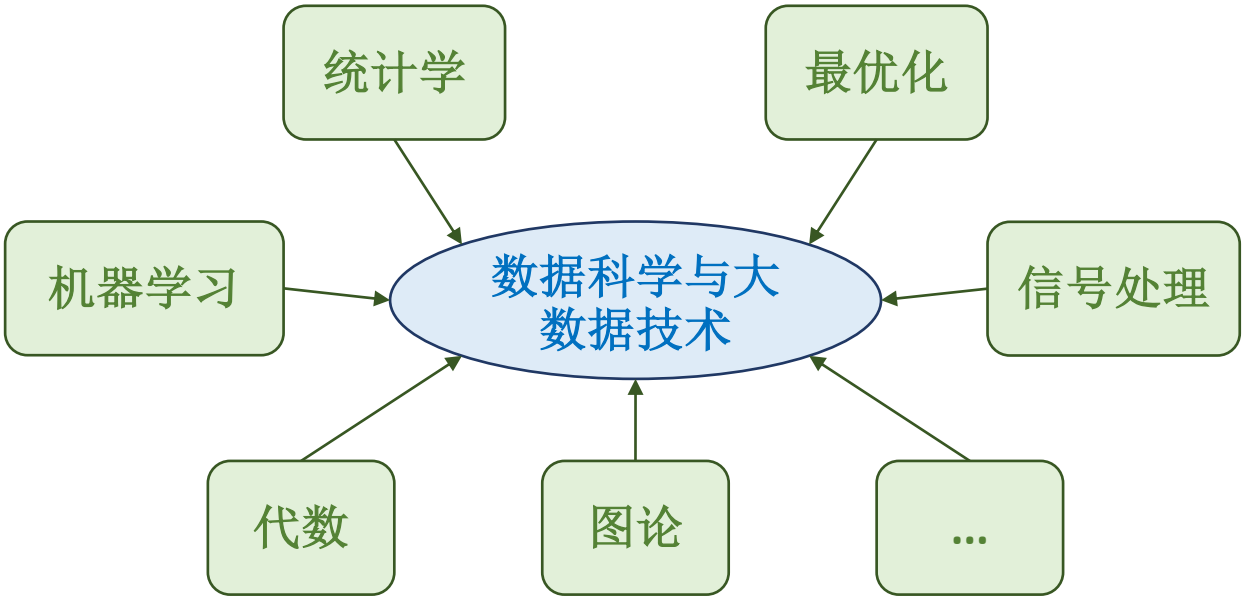
核磁共振成像（MRI）被广泛用于医疗诊断

4 如何在减少采样数目的情况下重建MRI图像，从而缩短MRI扫描时间



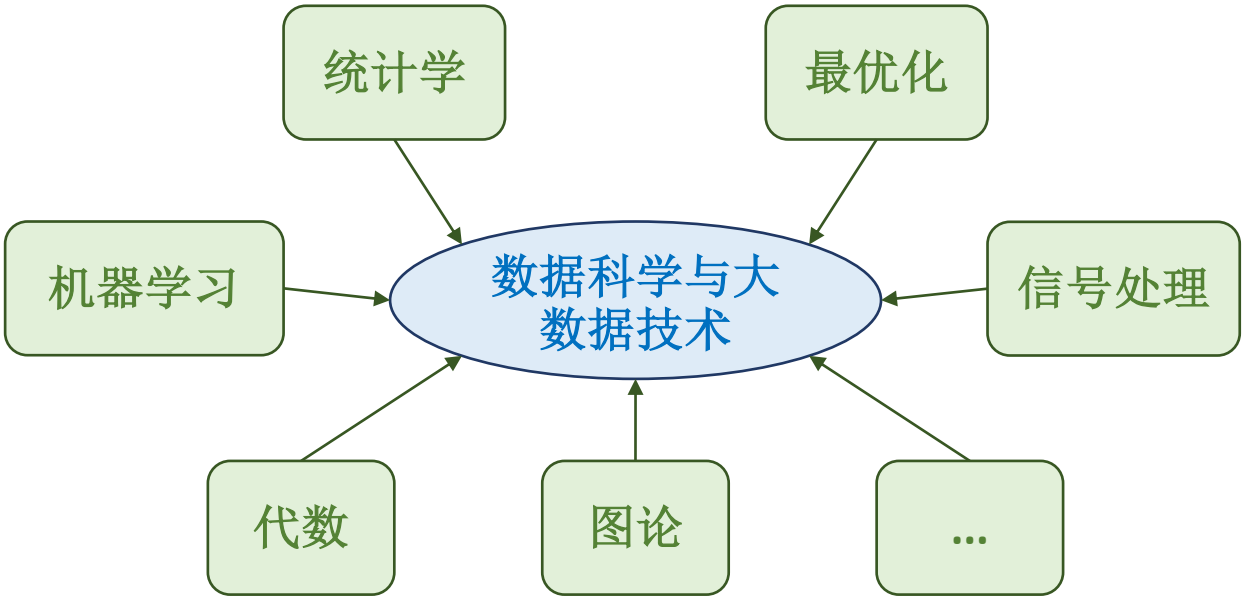
课程背景

为了对海量数据进行高效的存储/分析/…



课程背景

为了对海量数据进行高效的存储/分析/…



本课程：介绍一些存储/分析数据的常用数学工具和方法

课程内容

Part1 随机化方法	主要涉及概率论
一致性哈希 布隆过滤器 CM Sketch方法 最小哈希 欧氏距离下的相似搜索 Jaccard相似度下的相似搜索	
Part2 谱分析方法	主要涉及代数
主成分分析 奇异值分解 谱图论	
Part3 最优化方法	
压缩感知	

考核方式

- 5次乐学答题（5次×6分/次=30分）
  - 每次答题都有截止时间，一般9天时间作答

- 期末考试（70分）

* 请关注

i北理：发布各项通知/提醒

乐学平台：发布课件/答题

参考资料

- Stanford CS 168: The Modern Algorithmic Toolbox 课程讲义
  - by Tim Roughgarden and Gregory Valiant
- UMass Amherst COMPSCI 514: Algorithms for Data Science 课程课件
  - by Cameron Musco
- <Foundations of Data Science> 书籍
  - by Avrim Blum, John Hopcroft, and Ravindran Kannan
- ...

一致性哈希

分布式缓存

内容分发网络

为什么播放同样的视频，不同视频网站的加载速度可能差别较大?
为什么在同一个视频网络，播放不同（热度）的视频可能加载速度不同?

内容分发网络

为什么播放同样的视频，不同视频网站的加载速度可能差别较大?
为什么在同一个视频网络，播放不同（热度）的视频可能加载速度不同?

—— 缓存（caching）：当用户需要的文件已经存储在缓存服务器上，用户的请求可以得到快速响应

内容分发网络

比如在跨洋旅行/出差期间，想观看腾讯视频/爱奇艺/...
如果访问在北京的服务器上的文件，需要完成上万公里距离的文件传输，
这会带来高时延等问题

内容分发网络

比如在跨洋旅行/出差期间，想观看腾讯视频/爱奇艺/...
如果访问在北京的服务器上的文件，需要完成上万公里距离的文件传输，
这会带来高时延等问题

利用内容分发网络将热点文件存储在各地的缓存服务器上

内容分发网络

内容分发网络往往有多个层级的缓存服务器

当用户请求文件在下级服务器中未做缓存时，继续在上一级服务器中寻找（好比物流发货，先在本地仓库搜索库存，如果没有则在上级仓库搜索）

内容分发网络

内容分发网络往往有多个层级的缓存服务器

当用户请求文件在下级服务器中未做缓存时，继续在上一级服务器中寻找（好比物流发货，先在本地仓库搜索库存，如果没有则在上级仓库搜索）

图 45：京东物流设施布局（2020Q3）

数据来源：京东年报，西南证券整理

内容分发网络

互联网公司可以自行搭建内容分发网络，也可以将其外包

Akamai（阿卡迈）搭建的内容分发网络有部署在136个国家的约30万个边缘服务器，为Apple、Microsoft等公司提供缓存服务。服务全球网络15～30%的流量

内容分发网络

互联网公司可以自行搭建内容分发网络，也可以将其外包

Akamai（阿卡迈）搭建的内容分发网络有部署在136个国家的约30万个边缘服务器，为Apple、Microsoft等公司提供缓存服务。服务全球网络15～30%的流量

1997年，研发人员参加了在MIT举办的$50K创业大赛，凭借一致性哈希等技术进入前6名

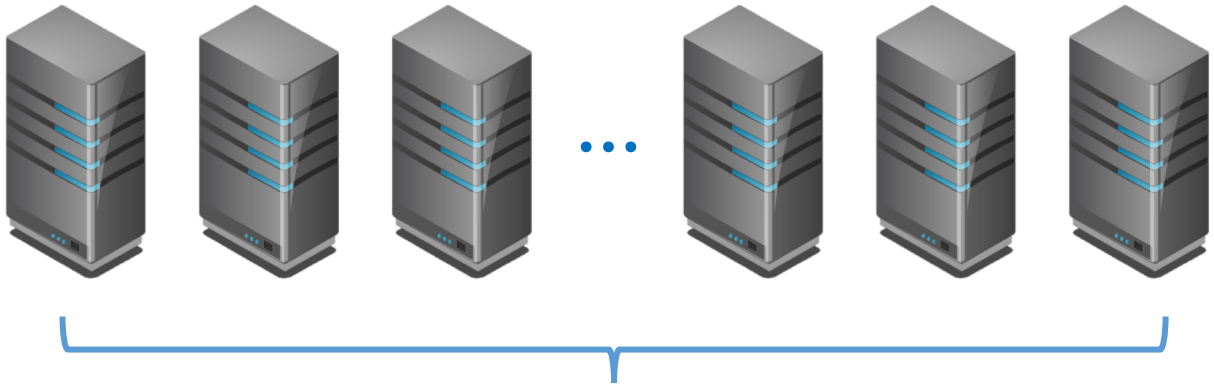
分布式缓存

在搭建边缘服务器时，一台机器无法满足大量本地用户的需求

分布式缓存

在搭建边缘服务器时，一台机器无法满足大量本地用户的需求

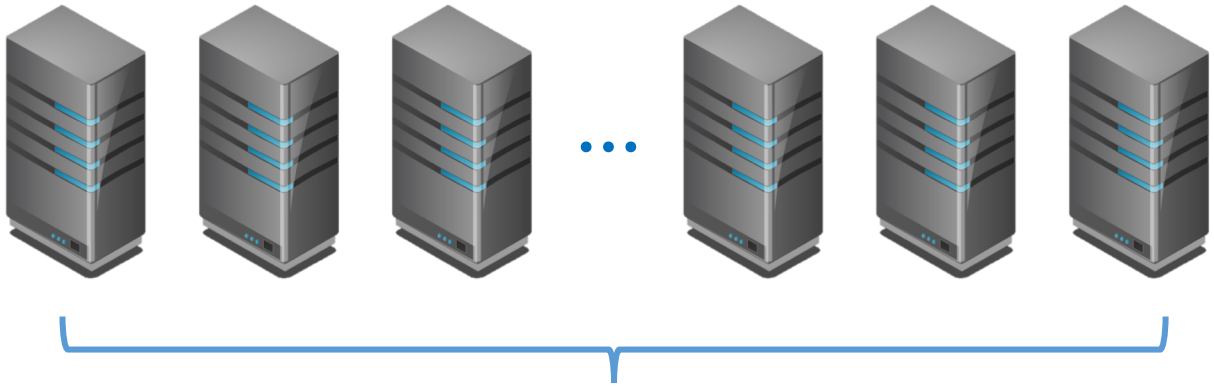
需要利用多台机器进行分布式缓存



分布式缓存

在搭建边缘服务器时，一台机器无法满足大量本地用户的需求

需要利用多台机器进行分布式缓存

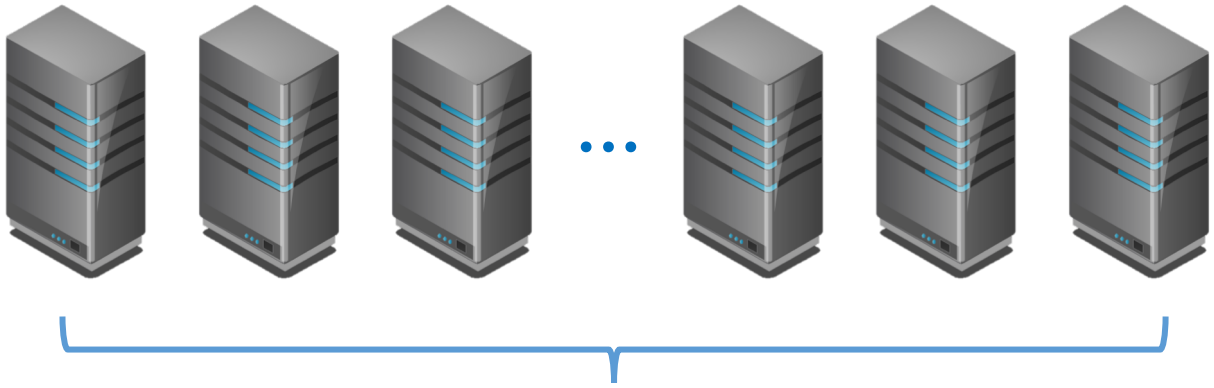


当用户需要访问文件，如何确定该文件是否已在这n台机器（如n=100）缓存？在哪一台机器上缓存？

分布式缓存

在搭建边缘服务器时，一台机器无法满足大量本地用户的需求

需要利用多台机器进行分布式缓存



当用户需要访问文件，如何确定该文件是否已在这n台机器（如n=100）缓存？在哪一台机器上缓存？

对所有机器依次逐个确认显然不是好方法

需要一种在给定文件名称时，可以迅速定位到相应机器的方法

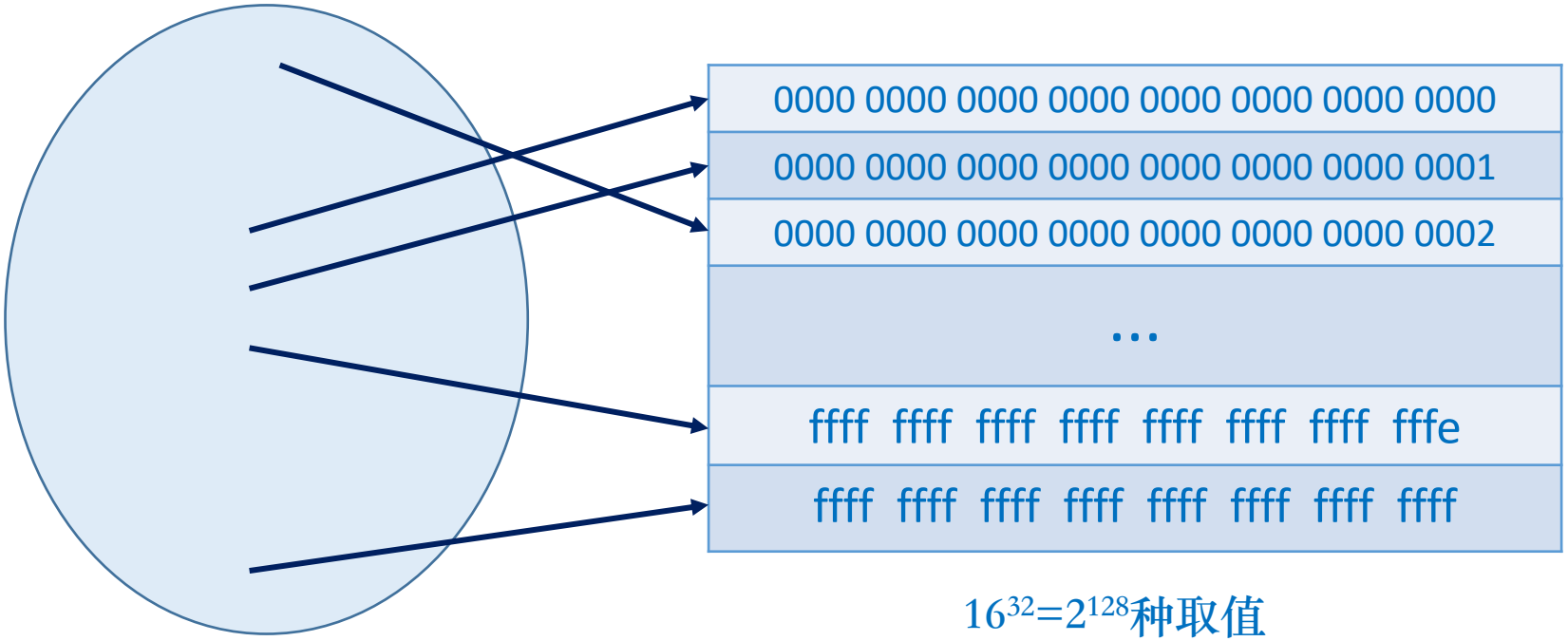
哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

回顾哈希函数



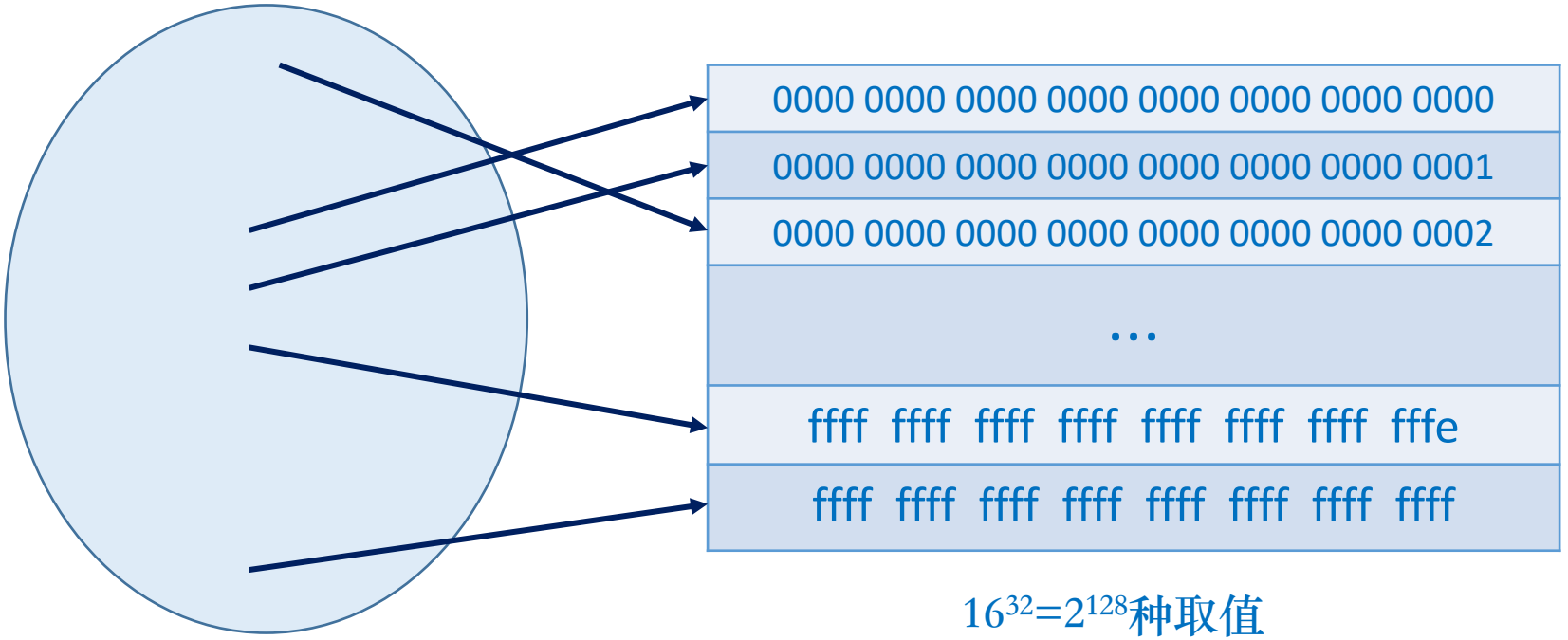
$16^{32}=2^{128}$种取值

比如URL的集合

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

回顾哈希函数



$16^{32}=2^{128}$种取值

比如URL的集合

“均匀地”将输入数据映射到不同数值

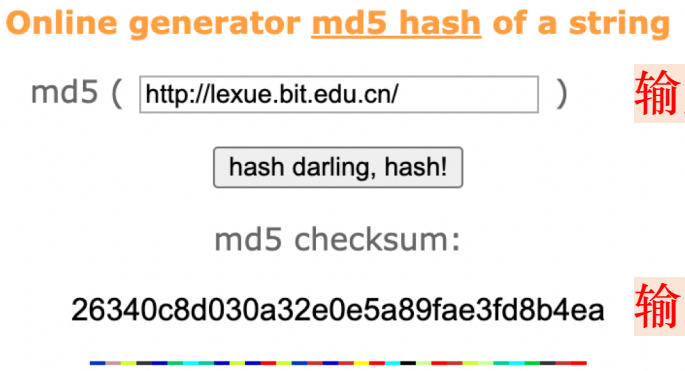
哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

回顾哈希函数

function md5()

MD5是常用的一种哈希函数



输入URL

输出32位（16进制）数值

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

回顾哈希函数

function md5()

MD5是常用的一种哈希函数

Online generator md5 hash of a string

md5 (http://lexue.bit.edu.cn/)

输入URL

hash darling, hash!

md5 checksum:

26340c8d030a32e0e5a89fae3fd8b4ea

输出32位（16进制）数值

如何根据得到的数值，确定文件/网页应该在$0,1,\ldots,n-1$中哪个机器缓存?

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

回顾哈希函数

function md5()

MD5是常用的一种哈希函数

Online generator md5 hash of a string

md5 (http://lexue.bit.edu.cn/)

输入URL

hash darling, hash!

md5 checksum:

26340c8d030a32e0e5a89fae3fd8b4ea

输出32位（16进制）数值

如何根据得到的数值，确定文件/网页应该在$0,1,\ldots,n-1$中哪个机器缓存?

数值对n取余，比如n=97时

26340c8d030a32e0e5a89fae3fd8b4ea mod n = 5

对n取余能保证结果一定在$\{0,1,\ldots,n-1\}$中取值

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件x的哈希值$h(x)$，然后计算$h(x) \bmod n$

存储文件时

缓存

提取文件时

请求

找到相应

完成文件提取

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件$\boldsymbol{x}$的哈希值$h(\boldsymbol{x})$，然后计算$h(\boldsymbol{x}) \bmod n$

为什么要用MD5算法？有什么很好的性质？

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件x的哈希值$h(x)$，然后计算$h(x) \bmod n$

为什么要用MD5算法？有什么很好的性质？

均匀分担存储任务

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件x的哈希值$h(x)$，然后计算$h(x) \bmod n$

当n动态变化（增加或减少）时，会有什么问题?

之前

之后

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件x的哈希值$h(x)$，然后计算$h(x) \bmod n$

当n动态变化（增加或减少）时，会有什么问题?

之前

之后

$h(x) \bmod n_1$ 很可能不等于$h(x) \bmod n_2$
即可能要变更所有文件的缓存位置

哈希函数

利用哈希函数建立从文件到机器编号（$0,1,\ldots,n-1$）的映射关系，再进行缓存/提取

方法：用MD5算法计算文件x的哈希值$h(x)$，然后计算$h(x) \bmod n$

当n动态变化（增加或减少）时，会有什么问题?

$h(x) \bmod n_1$ 很可能不等于$h(x) \bmod n_2$

如何在n动态变化时，保证大部分的文件不用变更存储的位置?

一致性哈希

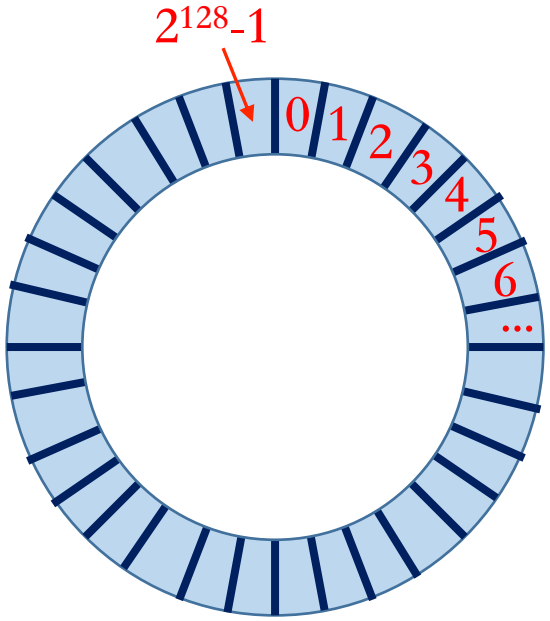
一致性哈希的方法

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值

一致性哈希

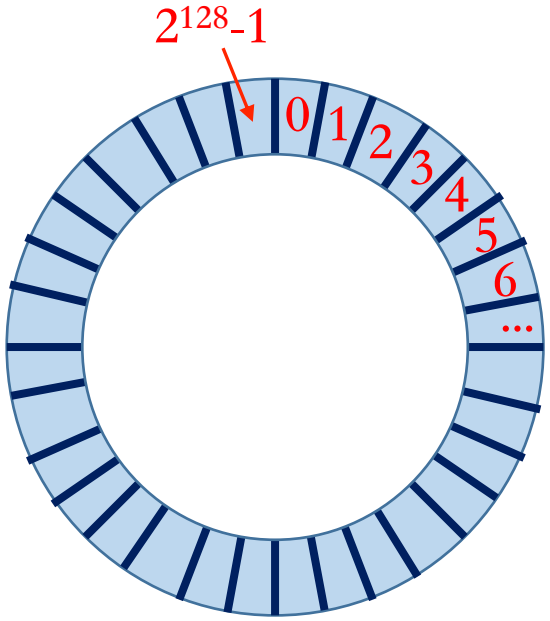
一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



原方法　如果当前有两台机器，那么$h(x)$为偶数的文件存储在第0台、奇数第1台

一致性哈希

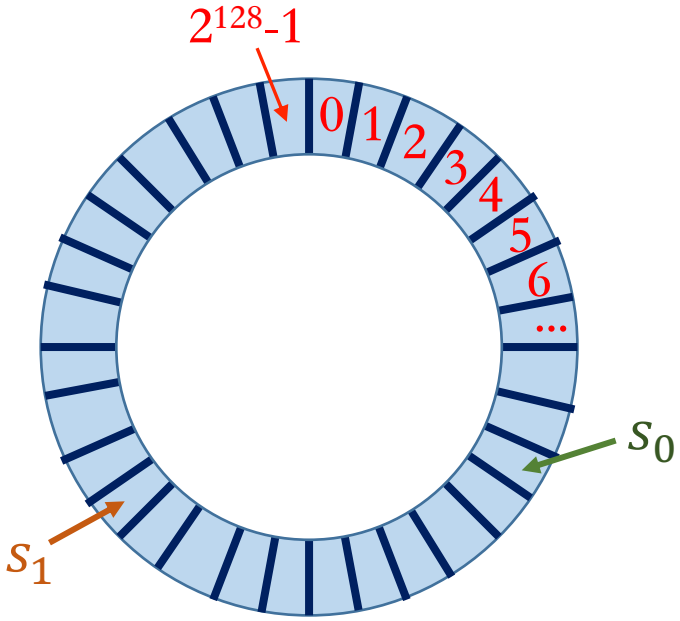
一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



一致性哈希　如果当前有两台机器，对应的哈希值为s_0和s_1

一致性哈希

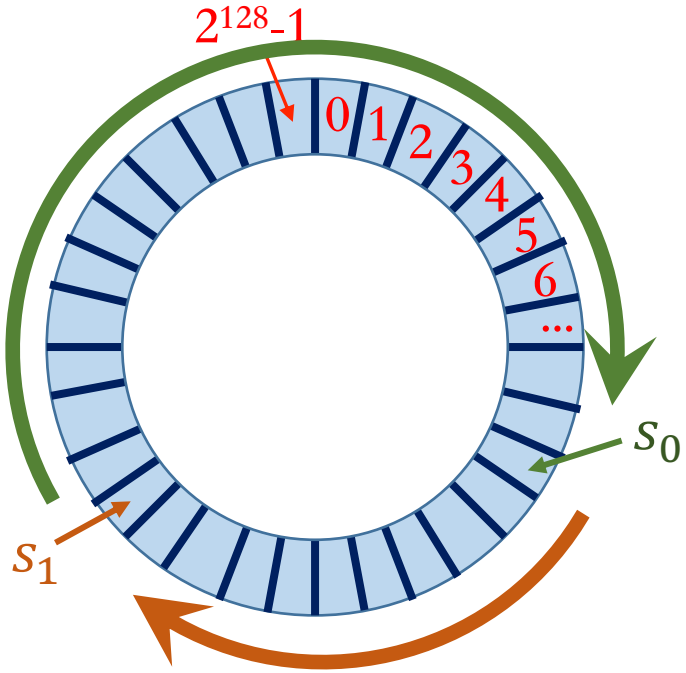
一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



一致性哈希　如果当前有两台机器，对应的哈希值为s_0和s_1

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



优点？

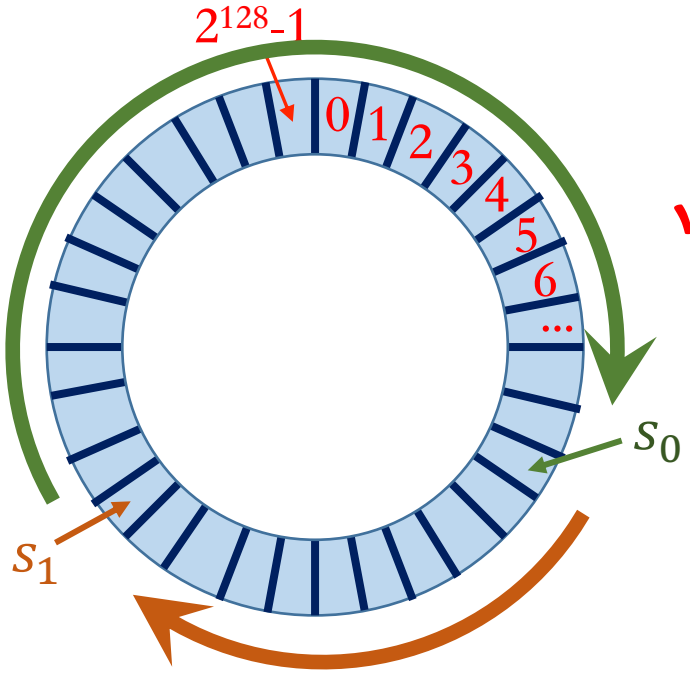
一致性哈希

如果当前有两台机器，对应的哈希值为s_0和s_1

对每一个文件x，从$h(x)$的位置出发沿顺时间找最接近的机器

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



优点？

✓ 依然保证文件相对均匀存储在n台机器上

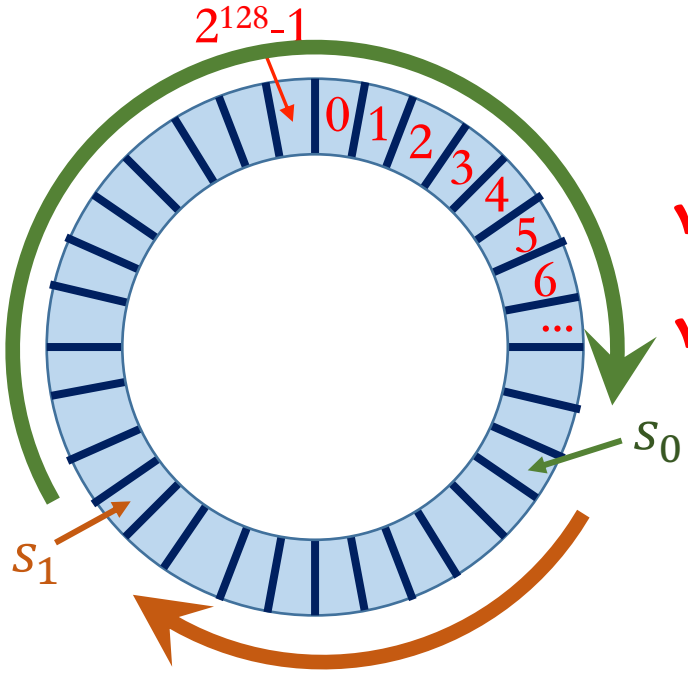
一致性哈希

如果当前有两台机器，对应的哈希值为s_0和s_1

对每一个文件x，从$h(x)$的位置出发沿顺时间找最接近的机器

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



优点？

✓ 依然保证文件相对均匀存储在n台机器上

✓ 当增减机器时，仅要移动少量文件

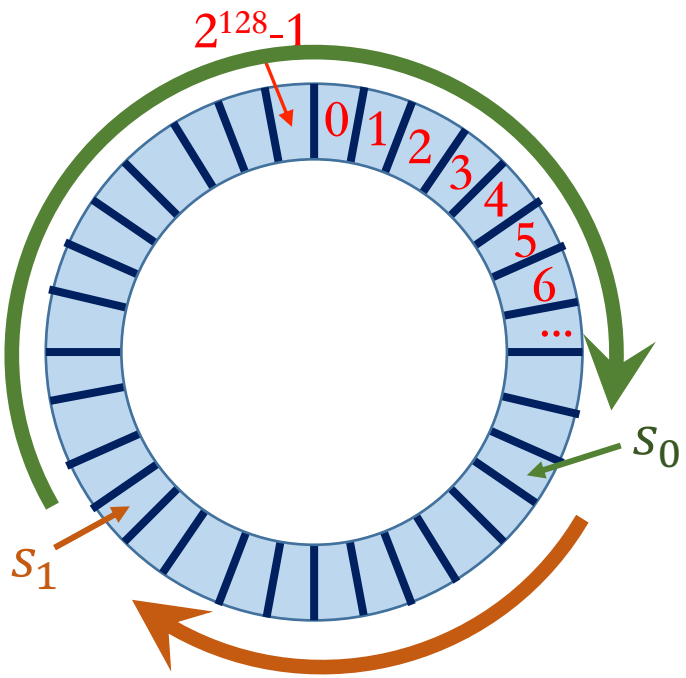
一致性哈希

如果当前有两台机器，对应的哈希值为s_0和s_1

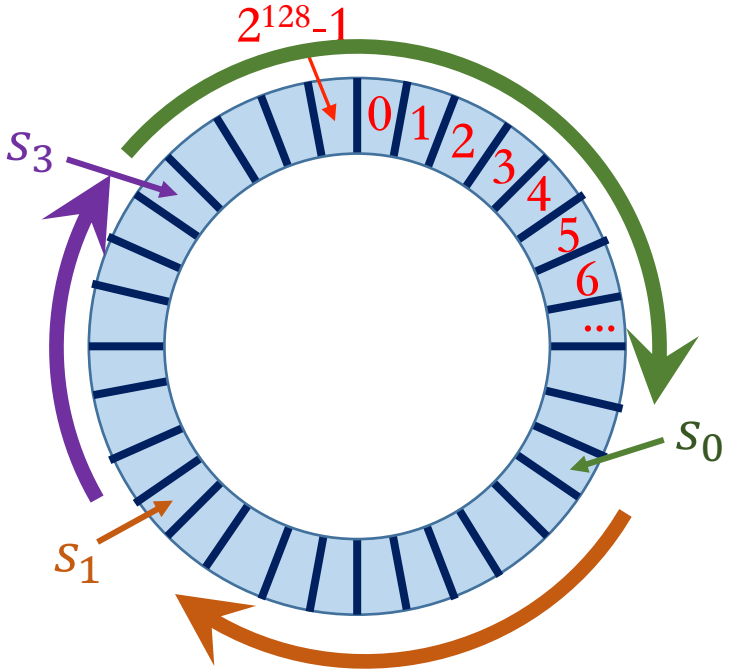
对每一个文件x，从$h(x)$的位置出发沿顺时间找最接近的机器

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



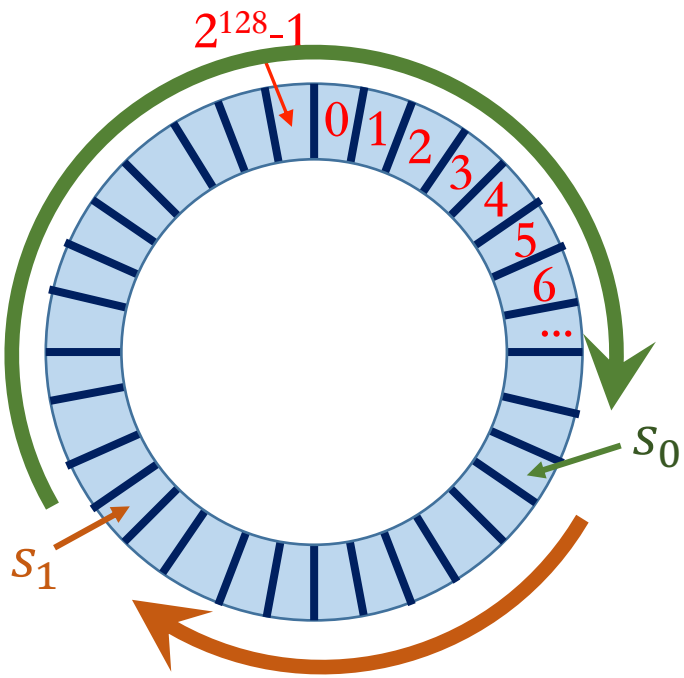
两台机器



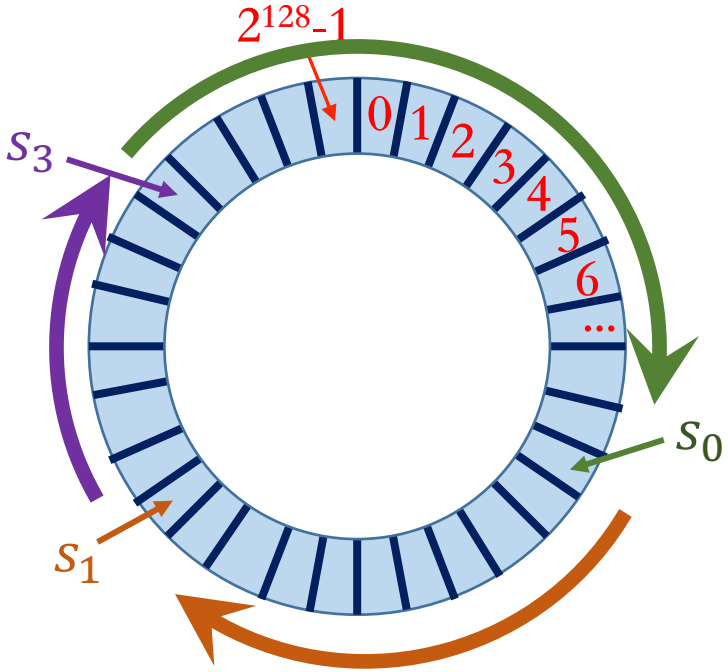
新增一台机器

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



两台机器

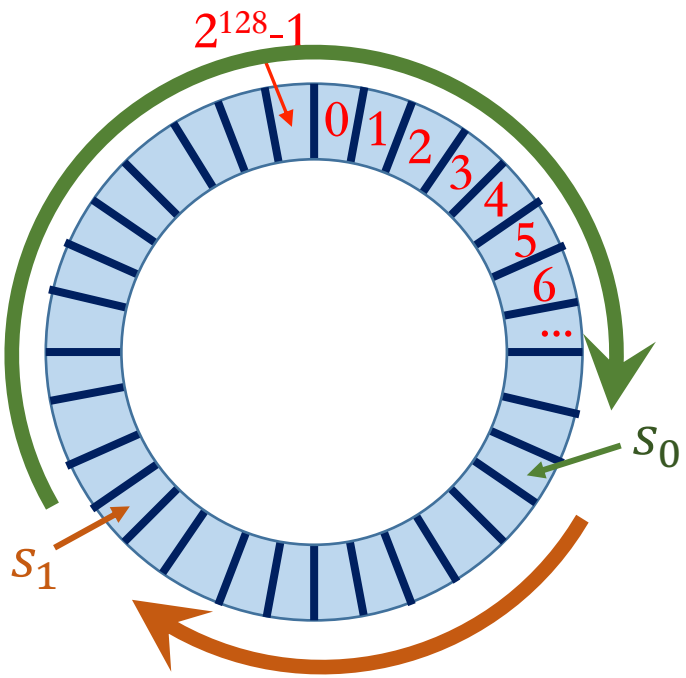


新增一台机器

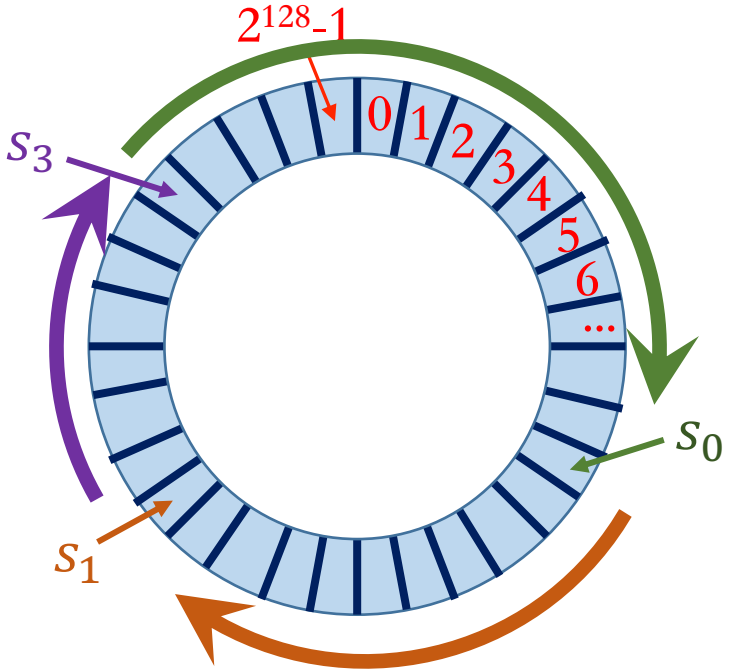
当机器数目从n-1增加到n时，平均需要移动的文件数是总文件数的 ?

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



两台机器



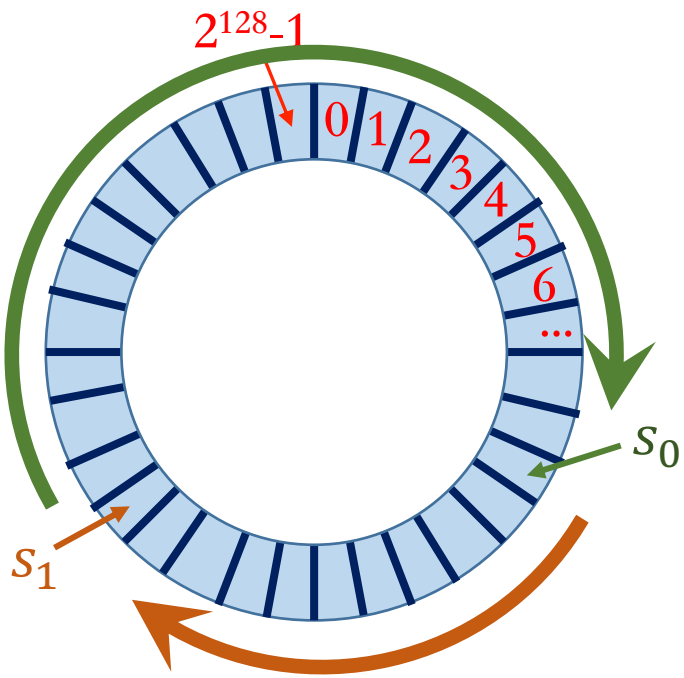
新增一台机器

考虑用最小哈希（见后续课件）的证明思路

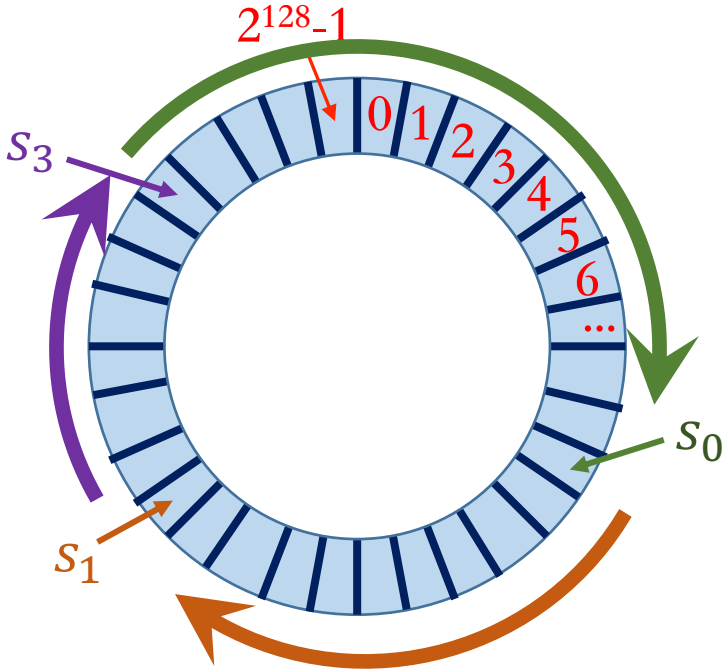
当机器数目从n-1增加到n时，平均需要移动的文件数是总文件数的1/n

一致性哈希

一致性哈希（consistent hashing）的解决思路：为每一台机器也计算一个哈希值



两台机器



新增一台机器

如果减少机器怎么办?

一致性哈希

以较大的概率，任何机器在“圆圈”上占据的区间比例都不会超过$O(\frac{logn}{n})$

$f(x) = O\big(g(x)\big)$ if there exists a positive real number M and a real number x_0 such that

$$|f(x)| \leq Mg(x) \quad \text{for all } x \geq x_0.$$

一致性哈希

以较大的概率，任何机器在“圆圈”上占据的区间比例都不会超过$O(\frac{logn}{n})$

- 将单位圆圈均匀切成$\frac{n}{\mathrm{c}\log n}$段

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n = e$$

- 某给定段不包含任何机器的概率：$\left(1-\frac{\mathrm{c}\log n}{n}\right)^n \approx e^{-\mathrm{c}\log n} = \frac{1}{n^c}$
- 利用布尔不等式，至少有一段不包含任何机器的概率≤ $\frac{n}{\mathrm{c}\log n}\times\frac{1}{n^c}$

“第i段不包含”与“第j段不包含”间不独立

Formally, for a countable set of events A_1, A_2, A_3, ..., we have

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mathbb{P}(A_i).$$

画图方便理解

- 所有段都包含至少一个机器的概率≥ $1-\frac{n}{\mathrm{c}\log n}\times\frac{1}{n^c}$
- 即至少以$1-\frac{n}{\mathrm{c}\log n}\times\frac{1}{n^c}$的概率，任何机器占据的区间比例都不会超过$\frac{2\mathrm{c}\log n}{n}$

一致性哈希

以较大的概率，任何机器在“圆圈”上占据的区间比例都不会超过$O(\frac{logn}{n})$

- 将单位圆圈均匀切成$\frac{n}{2\log n}$段
- 某给定段不包含任何机器的概率：$\left(1-\frac{2\log n}{n}\right)^n \approx e^{-2\log n} = \frac{1}{n^2}$
- 利用布尔不等式，至少有一段不包含任何机器的概率$\leq \frac{n}{2\log n} \times \frac{1}{n^2} \leq 1/n$

Formally, for a countable set of events A_1, A_2, A_3, ..., we have

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} \mathbb{P}(A_i).$$

画图方便理解

- 所有段都包含至少一个机器的概率$\geq 1-\frac{1}{n}$
- 即至少以$1-\frac{1}{n}$的概率，任何机器占据的区间比例都不会超过$\frac{4\log n}{n}$

一致性哈希

以较大的概率，至少一个机器在“圆圈”上占据的区间比例小于$\frac{1}{n^2}$

一致性哈希

以较大的概率，至少一个机器在“圆圈”上占据的区间比例小于$\frac{1}{n^2}$

In probability theory, the **birthday problem** asks for the probability that, in a set of n randomly chosen people, at least two will share a birthday. The **birthday paradox** is that, counterintuitively, the probability of a shared birthday exceeds 50% in a group of only 23 people.

假设有m（m>n）个大小均等段，每个段都最多包含一个机器的概率?

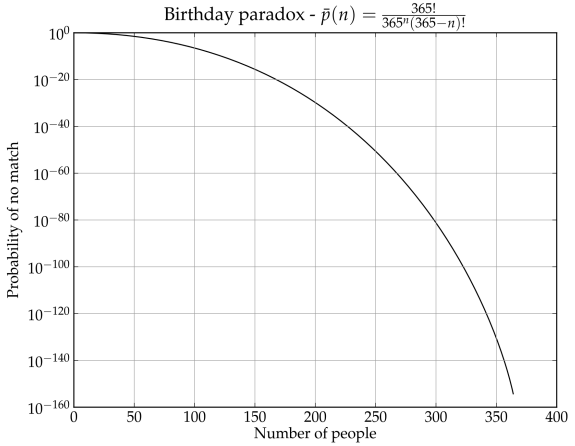
一致性哈希

以较大的概率，至少一个机器在“圆圈”上占据的区间比例小于$\frac{1}{n^2}$

In probability theory, the **birthday problem** asks for the probability that, in a set of n randomly chosen people, at least two will share a birthday. The **birthday paradox** is that, counterintuitively, the probability of a shared birthday exceeds 50% in a group of only 23 people.

假设有m（m>n）个大小均等段，每个段都最多包含一个机器的概率

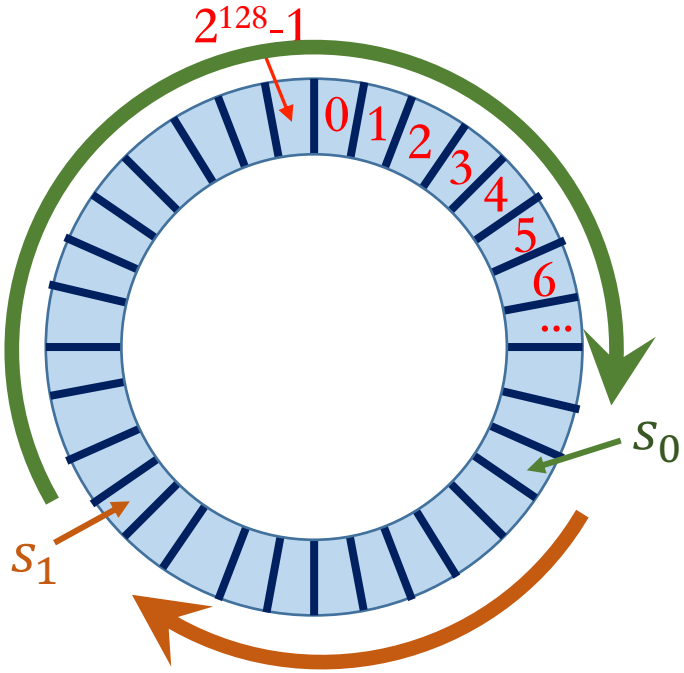
$$\frac{m(m-1)\cdots(m-n+1)}{m^n}$$



引自Wikipedia “Birthday problem”词条

一致性哈希

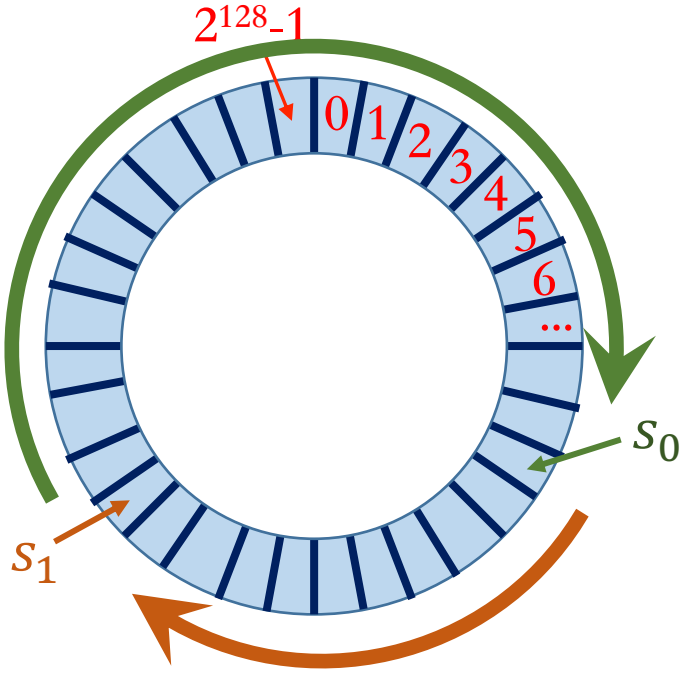
改进：如何保证文件“更均匀”地存储在n台机器上?



两台机器

一致性哈希

改进：如何保证文件“更均匀”地存储在n台机器上?

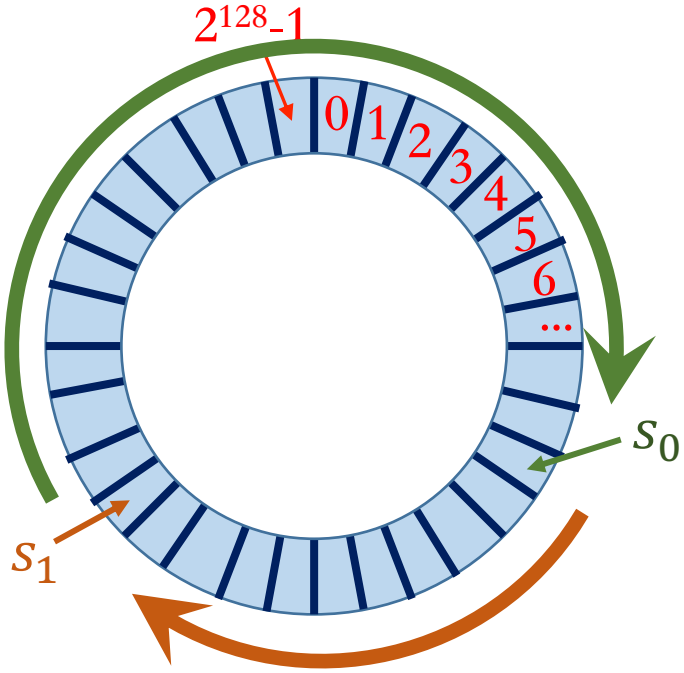


两台机器

假设有m个文件，每台机器存储的文件数是一个随机变量，其期望值为 ?

一致性哈希

改进：如何保证文件“更均匀”地存储在n台机器上?



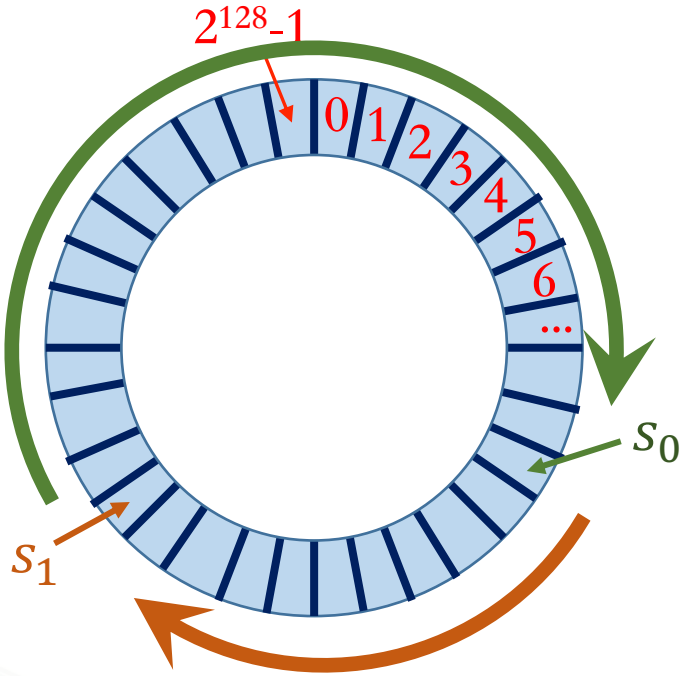
两台机器

假设有m个文件，每台机器存储的文件数是一个随机变量，其期望值为m/2，但是方差较大

一致性哈希

改进：如何保证文件“更均匀”地存储在n台机器上?

解决方法：利用多个不同的哈希函数，为每台机器计算多个映射值

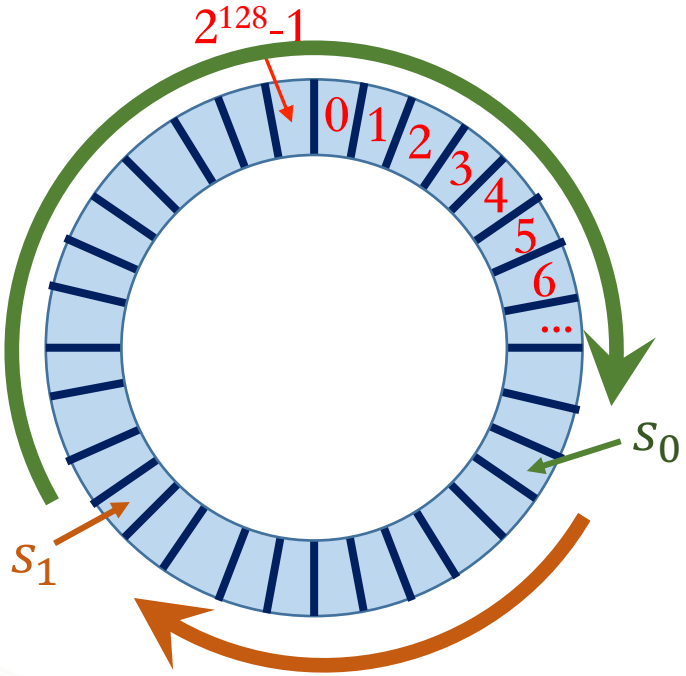


两台机器

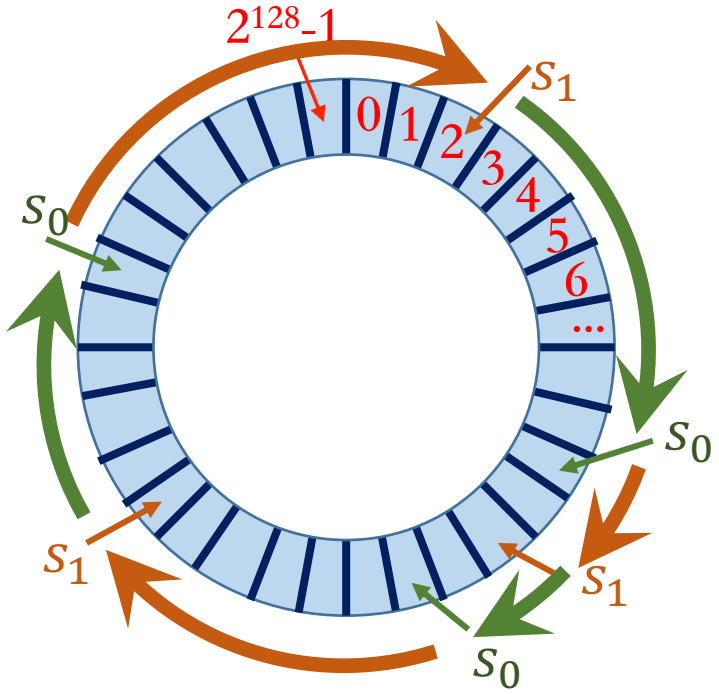
一致性哈希

改进：如何保证文件“更均匀”地存储在n台机器上?

解决方法：利用多个不同的哈希函数，为每台机器计算多个映射值



两台机器

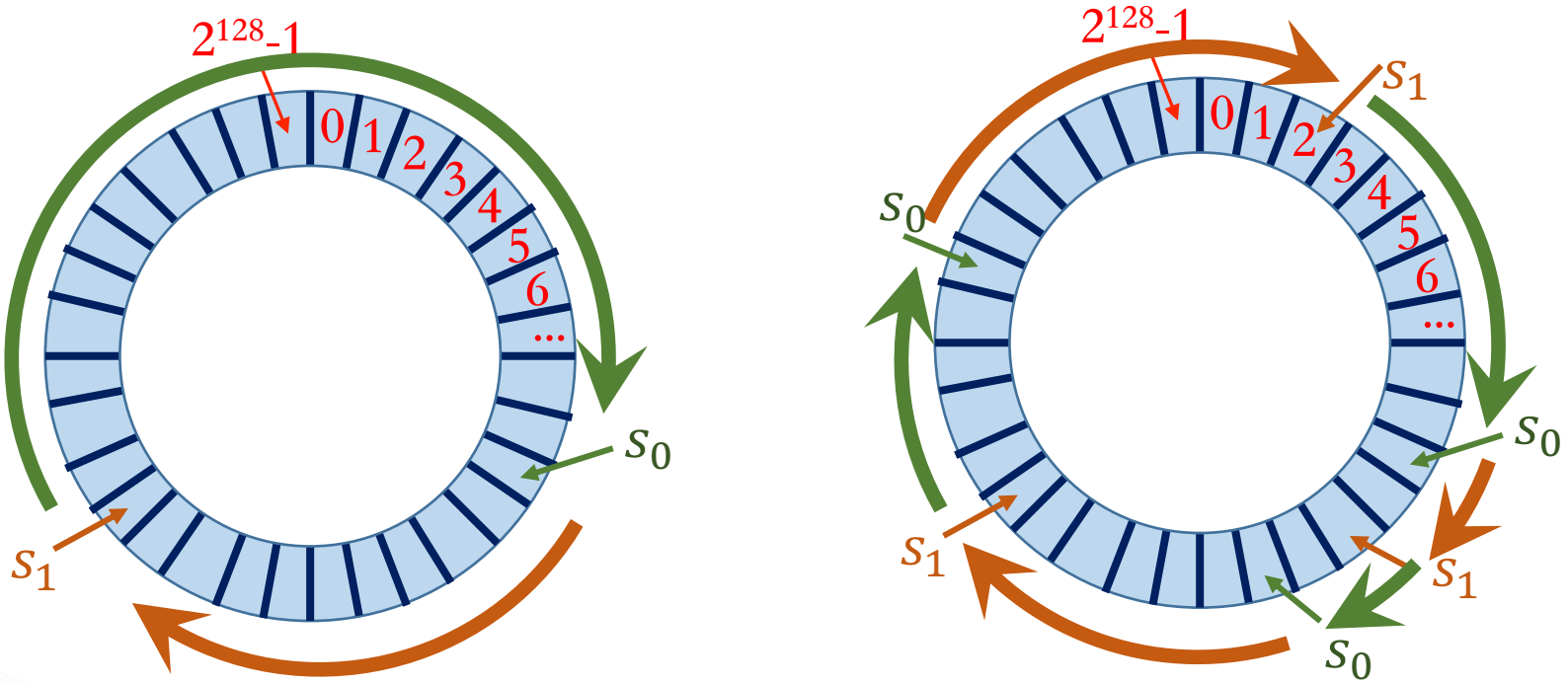


两台机器

一致性哈希

改进：如何保证文件“更均匀”地存储在n台机器上?

解决方法：利用多个不同的哈希函数，为每台机器计算多个映射值



有一个附加优势，当机器的存储空间大小有差异时?

一致性哈希

一致性哈希的历史

相关历史

- 1997年，MIT研究人员于STOC发表相关论文

通过二叉树完成对文件/机器的搜索

[PDF] Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web

D Karger, E Lehman, T Leighton, R Panigrahy… - Proceedings of the …, 1997 - dl.acm.org

Wedescribe a family of caching protocols for distrib-uted networks that can be used to decrease or eliminate the occurrence of hot spots in the network. Our protocols are particularly designed for use with very large networks such as the Internet, where delays caused by hot spots can be severe, and where it is not feasible for every server to have complete information about the current state of the entire network. The protocols are easy to implement using existing network protocols such as TCP/fF', and require very little overhead …

☆ Save 99 Cite Cited by 2965 Related articles All 59 versions

相关历史

- 1997年，MIT研究人员于STOC发表相关论文

[PDF] Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web

D Karger, E Lehman, T Leighton, R Panigrahy… - Proceedings of the …, 1997 - dl.acm.org

Wedescribe a family of caching protocols for distrib-uted networks that can be used to decrease or eliminate the occurrence of hot spots in the network. Our protocols are particularly designed for use with very large networks such as the Internet, where delays caused by hot spots can be severe, and where it is not feasible for every server to have complete information about the current state of the entire network. The protocols are easy to implement using existing network protocols such as TCP/fF', and require very little overhead …

☆ Save 99 Cite Cited by 2965 Related articles All 59 versions

[1]Laboratory for Computer Science, MIT, Cambridge, MA 02139.
email: {karger,e_lehman,danl,ftl,mslevine,danl,rinap}@theory.lcs.mit.edu
A full version of this paper is availble at:
http://theory.lcs.mit.edu/~ {karger,e_lehman,ftl,mslevine,danl,rinap}
[2]Department of Mathematics, MIT, Cambridge, MA 02139

STOC '97 El Paso, Texas USA

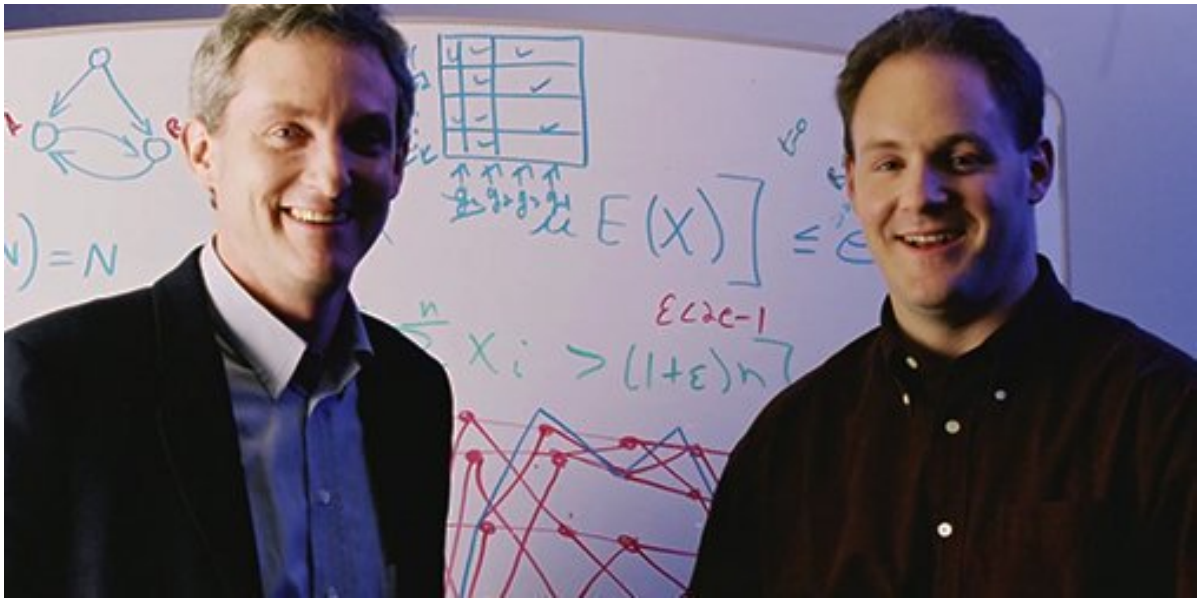
计算机科学理论领域三大会议

- STOC（ACM Symposium on Theory of Computing）
- SODA（ACM-SIAM Symposium on Discrete Algorithms）
- FOCS（IEEE Annual Symposium on Foundations of Computer Science）

（曾因被认为难以应用于实际而拒稿）

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立



Tom Leighton与Danny Lewin

Akamai创始人Tom Leighton介绍公司成立的故事
https://www.akamai.com/company/company-history?

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立

1995年，万维网之父Tim Berners-Lee在MIT提出缓解网络拥塞的挑战

研究并行算法与结构的MIT应用数学教授Tom Leighton意识到利用并行算法可能可以解决该问题

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立

1995年，万维网之父Tim Berners-Lee在MIT提出缓解网络拥塞的挑战

研究并行算法与结构的MIT应用数学教授Tom Leighton意识到利用并行算法可能可以解决该问题

1996年，在以色列取得学士学位的Danny Lewin到MIT与Tom Leighton一同工作，团队提出利用分布式机器解决拥塞问题的方案

1997年，Tom Leighton与Danny Lewin开始考虑算法的商用问题。在与MIT斯隆商学院学生Preetish Nijhawan的合作下，团队参加了为期9个月的MIT $50K创业大赛，入围大奖候选名单（6/100）

1998年，Tom Leighton与Danny Lewin创办Akamai，大部分员工是参与项目的MIT学生

...

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立
- 1999年3月31日，星球大战I预告片发布，其余网站迅速崩溃，只有Akamai支持的两个网站经受住冲击，因此声名鹊起

In March 1999, the technology proved itself. Between Entertainment Tonight's website hosting Phantom Menace trailers and ESPN's online coverage of March Madness, Akamai successfully handled 250 million hits on those two sites alone. While other websites covering the same ground crashed under the strain, Akamai's technology juggled as much as 3,000 hits per second.

引自Richard Trenholm, CNET

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立
- 1999年3月31日，星球大战I预告片发布，其余网站（包括Apple.com）迅速崩溃，只有Akamai支持的两个网站经受住冲击，因此声名鹊起
- 1999年4月1日，Steve Jobs致电Akamai总裁Paul Sagan，乌龙事件

The next day, Jobs put in a call to Paul Sagan, who was then the president of Akamai.

Exactly what they each said is unclear, but at some point in the conversation Jobs did mention the idea of acquiring Akamai, according to Leighton.

But Sagan hung up the phone mid-conversation.

Why? Because Jobs was calling on April 1 — and Sagan thought the call was an April Fool's joke.

Sagan simply didn't believe he was actually speaking with Steve Jobs.

What happened after that: Akamai launched its service commercially later that month, and Apple did go on to buy a piece of Akamai that June — a 5 percent stake, worth $12.5 million.

引自Kyle Alspach, Boston Business Journal

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立
- 1999年3月31日，星球大战I预告片发布，其余网站（包括Apple.com）迅速崩溃，只有Akamai支持的两个网站经受住冲击，因此声名鹊起
- 1999年4月1日，Steve Jobs致电Akamai总裁Paul Sagan，乌龙事件
- 2001年9月11日，创始人之一Danny Lewin在与“911恐怖袭击”劫机者搏斗中死亡

以Danny Lewin命名的公园

with them almost since the beginning. The next morning Lewin had to fly from Boston to Los Angeles.

"He probably barely got an hour of sleep before getting on board the next morning," Leighton remembers.

Lewin was sitting in seat 9B. With his Israeli military training and understanding of Arabic, he may have figured out what was going on, perhaps even tried to stop it. According to flight attendants' calls relayed to authorities on the ground, the first passenger to be killed was seated in 9B. He was stabbed to death.

引自Todd Leopold, CNN

相关历史

- 1997年，MIT研究人员于STOC发表相关论文
- 1998年，Akamai建立
- 1999年3月31日，星球大战I预告片发布，其余网站（包括Apple.com）迅速崩溃，只有Akamai支持的两个网站经受住冲击，因此声名鹊起
- 1999年4月1日，Steve Jobs致电Akamai总裁Paul Sagan，乌龙事件
- 2001年9月11日，创始人之一Danny Lewin在与“911恐怖袭击”劫机者搏斗中死亡
- 2001年，一致性哈希被用于解决P2P网络（如BitTorrent协议）中文件搜索的问题

Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

Ion Stoica; Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan†
MIT Laboratory for Computer Science
chord@lcs.mit.edu
http://pdos.lcs.mit.edu/chord/

该改工作引用量>18000

SIGCOMM'01, August 27-31, 2001, San Diego, California, USA.

SIGCOMM是计算机网络领域的最高级别会议

本讲小结

- 分布式缓存
- 一致性哈希的方法与历史

计算机学院
余皓然

主要参考资料

Tim Roughgarden and Gregory Valiant <CS 168 - The Modern Algorithmic Toolbox> Lecture Notes

Cameron Musco <COMPSCI 514 - Algorithms for Data Science> Slides

Ankur Moitra <6.854/18.415 Advanced Algorithms> Video

谢谢！

计算机学院
余皓然
2023/4/18